

(TRANSLATION)

Our Ref.: OP1439-US

Prior Art Reference 2:

Japanese Patent Laid-Open Publication No. 63(1988)-211433

Laid-Open Date: September 2, 1988

Title of the Invention: PRIOR ERROR CHECK SYSTEM FOR GENERATING
COMMAND PROCEDURE/JOB CONTROL LANGUAGE

Patent Application No. 62(1987)-44902

Filing Date: February 27, 1987

Inventor: Naoko HASHIMOTO
c/o Nippon Denki Kabushiki Kaisha (NEC CORP.)
Minato-ku, Tokyo, Japan

Applicant: NIPPON DENKI KABUSHIKI KAISHA (NEC CORP)
Minato-ku, Tokyo, Japan

- - - - -
Translation of Claim
(this cited document has only one claim)

Claim:

A prior error check system for generating command procedure/job
control language, comprising:

flow chart generating means for generating command procedure or
a flow of the job control language by displaying the flow chart on
a terminal; and

parameter input means capable of inputting the necessary parameter
by displaying on the terminal a parameter input screen corresponding
to the instructed function; the system further comprising:

parameter analyzing means for recognizing the inputted parameter
by analyzing the command or the job control language which is directly

described on the flow chart displayed on the terminal;

parameter conversion means for generating a parameter table to perform an error check based on the parameter which has been recognized by the parameter analyzing means or the parameter which has been inputted by the parameter input means;

error detection means for detecting a parameter error by checking the contents of the parameter table based on the data for the parameter check; and

error notifying means for displaying on the terminal the error contents detected by the error detection means.

/ / / / / / / / / / LAST ITEM / / / / / / / / / /

PATENT ABSTRACTS OF JAPAN

(11)Publication number : 63-211433

(43)Date of publication of application : 02.09.1988

(51)Int.Cl.

G06F 9/06

G06F 3/02

G06F 11/28

(21)Application number : 62-044902

(71)Applicant : NEC CORP

(22)Date of filing : 27.02.1987

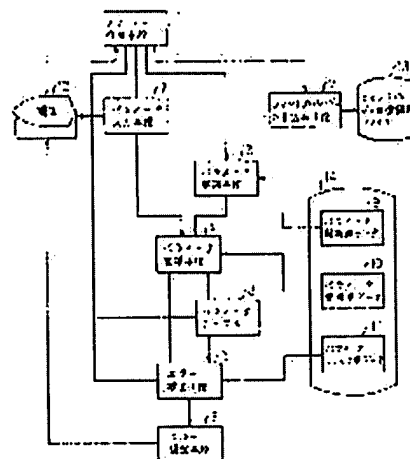
(72)Inventor : HASHIMOTO NAOKO

(54) PRIOR ERROR CHECK SYSTEM FOR GENERATING COMMAND PROCEDURE/JOB CONTROL LANGUAGE

(57)Abstract:

PURPOSE: To generate correct command procedure and job control language by checking a parameter designated to a command and a job control language, and displaying an error message, if an error is detected.

CONSTITUTION: When an input of a parameter is ended in accordance with a parameter input screen, what kind of parameter has been inputted is informed from a parameter input means 2 to a parameter converting means 4. Also, when it is ended to designate directly the parameter onto a flow chart, what kind of parameter has been inputted is analyzed by a parameter analyzing means 3, and by the parameter converting means 4, a parameter table 8 for checking the parameter is generated at every command. Subsequently, by an error detecting means 5, the contents of the parameter table 8 are examined, and when a parameter error exists, the contents of the error are displayed on a terminal 12 by an error informing means 6. In such a way, correct command procedure and job control language can be generated.



LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's
decision of rejection]

[Date of extinction of right]

⑫ 公開特許公報(A)

昭63-211433

⑤ Int. Cl.⁴G 06 F 9/06
3/02
11/28

識別記号

3 2 0
3 8 0
3 0 5

庁内整理番号

N-7361-5B
C-8724-5B
7343-5B

④ 公開 昭和63年(1988)9月2日

審査請求 未請求 発明の数 1 (全9頁)

⑥ 発明の名称 コマンドプロシージャ/ジョブ制御言語の作成における事前エラー
チェック方式

⑦ 特 願 昭62-44902

⑧ 出 願 昭62(1987)2月27日

⑨ 発 明 者 橋 本 直 子 東京都港区芝5丁目33番1号 日本電気株式会社内

⑩ 出 願 人 日本電気株式会社 東京都港区芝5丁目33番1号

⑪ 代 理 人 弁理士 境 廣 巳

明 細 書

1. 発明の名称

コマンドプロシージャ/ジョブ制御言語
の作成における事前エラーチェック方式

2. 特許請求の範囲

フローチャートを端末上に表示しコマンドプロシージャ或いはジョブ制御言語の読れを作成するフローチャート作成手段と、指示された機能に対応するパラメータ入力画面を端末上に表示し必要なパラメータを入力できるパラメータ入力手段とを有するコマンドプロシージャ或いはジョブ制御言語の作成システムにおいて、

前記端末上に表示されたフローチャート上で直接に記述されたコマンド或いはジョブ制御言語を解析し、入力されたパラメータを認識するパラメータ解析手段と、

該パラメータ解析手段により認識されたパラメータ、或いは前記パラメータ入力手段によって入力されたパラメータに基づいて、エラーチェックを行なうためのパラメータテーブルを作成するパ

ラメータ変換手段と、

前記パラメータテーブルの内容をパラメータチェック用データに基づいてチェックしてパラメータエラーを検出するエラー検出手段と、

該エラー検出手段により検出されたエラー内容を前記端末上に表示するエラー通知手段とを具備したことを特徴とするコマンドプロシージャ/ジョブ制御言語の作成における事前エラーチェック方式。

3. 発明の詳細な説明

(産業上の利用分野)

本発明は、コマンドプロシージャやジョブ制御言語の作成における事前エラーチェック方式に関する。

(従来の技術)

コマンドプロシージャやジョブ制御言語の作成を容易にする為に、フローチャートを端末上に表示しコマンドプロシージャやジョブ制御言語の読れを作成するフローチャート作成手段を用いて対話形式でコマンドプロシージャやジョブ制御言語

を作成するシステムが知られている。

また、コマンドやジョブ制御言語に与えるパラメータの入力を容易にする為に、指示された機能に対応した設定すべきパラメータ名等を含むパラメータ入力画面を端末上に表示し、必要なパラメータを逐次入力できるパラメータ入力手段を備えたシステムも知られている。

上述のようなシステムにおいては、コマンドプロシージャやジョブ制御言語の記述方法および形式については、フローチャートを利用することにより、また内容となるコマンドやジョブ制御言語についてはパラメータ入力画面を利用することによって、それぞれ詳細な知識がなくても、比較的簡単に目的に合ったコマンドプロシージャやジョブ制御言語が作成できる。

ところで、コマンドやジョブ制御言語中に指定するパラメータには各種のものが有り、必ず指定しなければならない必須パラメータや、或るパラメータを指定した場合には別の或るパラメータを必ず指定しなければならない必須関係のパラメー

た。また登録されたコマンドプロシージャ/ジョブ制御言語を実際に起動してみなければエラーが発見されず、作成段階でエラーの有無を作成者が知ることはできなかったもので、登録されたコマンドプロシージャやジョブ制御言語の再作成が必要となり、目的に合ったコマンドプロシージャやジョブ制御言語を完成するまでに長時間を要する欠点もあった。

本発明はこのような従来の問題点を解決したもので、その目的は、コマンドやジョブ制御言語に指定されるパラメータをそのコマンドやジョブ制御言語の作成段階でチェックし、若し誤りがあれば作成者に直ちに通知して修正を促すコマンドプロシージャ/ジョブ制御言語の作成における事前エラーチェック方式を提供することにある。

(問題点を解決するための手段)

本発明は上記目的を達成するために、フローチャートを端末上に表示しコマンドプロシージャ或いはジョブ制御言語の流れを作成するフローチャート作成手段と、指示された機能に対応するパラ

メータ群若しくはその反対に或るパラメータを指定したときは別の或るパラメータは指定できない排他関係のパラメータ群や、更に指定値の字數制限など各種のパラメータ指定規則が存在する。

上述したフローチャート作成手段、パラメータ入力手段は上述のようなパラメータ指定規則の個々についてまで作成者に教示するものではなく、この為、パラメータ指定規則に違反したパラメータがしばしば入力される。そしてこのような場合、従来は、指定されたパラメータの内容に関するチェックは行なっていないので、誤ったパラメータを含むコマンドプロシージャやジョブ制御言語が登録用ファイルに登録されてしまっていた。

(発明が解決しようとする問題点)

上述したように、従来は、コマンドやジョブ制御言語に指定したパラメータに関するチェックは行なわれていなかった為、エラーを有するパラメータを含むコマンドプロシージャ/ジョブ制御言語が登録用ファイルに格納される可能性が高く、従ってその実行中にエラーを発生する場合も多か

った。また登録されたコマンドプロシージャ/ジョブ制御言語を実際に起動してみなければエラーが発見されず、作成段階でエラーの有無を作成者が知ることはできなかったもので、登録されたコマンドプロシージャやジョブ制御言語の再作成が必要となり、目的に合ったコマンドプロシージャやジョブ制御言語を完成するまでに長時間を要する欠点もあった。

前記端末上に表示されたフローチャート上で直接に記述されたコマンド或いはジョブ制御言語を解析し、入力されたパラメータを認識するパラメータ解析手段と、

該パラメータ解析手段により認識されたパラメータ、或いは前記パラメータ入力手段によって入力されたパラメータに基づいて、エラーチェックを行なうためのパラメータテーブルを作成するパラメータ変換手段と、

前記パラメータテーブルの内容をパラメータチェック用データに基づいてチェックしてパラメータエラーを検出するエラー検出手段と、

該エラー検出手段により検出されたエラー内容を前記端末上に表示するエラー通知手段とを備える。

(作用)

パラメータ入力手段によって端末に表示されたパラメータ入力画面に従ってパラメータを入力し終えると、どのようなパラメータが入力されたかがパラメータ入力手段からパラメータ変換手段に通知され、またパラメータ入力画面を使用せずフローチャート上に直接にパラメータを指定し終えると、パラメータ解析手段でどのようなパラメータが入力されたかが解析されてパラメータ変換手段に通知され、パラメータ変換手段によってパラメータエラーチェック用のパラメータテーブルが、各コマンド毎に作成される。そして、エラー検出手段によって、上記パラメータテーブルの内容がパラメータチェック用データを参照して吟味され、パラメータエラーが存在するときは、エラー通知手段によってエラー内容が端末に表示される。

(実施例)

次に本発明の実施例について図面を参照して説明する。

第1図は本発明の実施例のブロック図であり、フローチャート作成手段1と、パラメータ入力手

持っている。

第1図において、コマンドプロシージャの作成に際し、作成者が端末12から所定の指示を与えると、フローチャート作成手段1が起動され、フローチャート作成手段1は端末12へのフローチャートの表示を開始し、作成者がコマンド等を入力することにより、フローチャート作成手段1は逐次コマンドプロシージャの処理の流れを表すフローチャートを作成して行く。

コマンドプロシージャの作成処理の一環として、あるコマンドをパラメータを含めて作成者が端末画面のフローチャート上に直接に記述すると、その内容はフローチャート作成手段1を介してパラメータ解析手段3に与えられ、パラメータ解析手段3はファイル14中のパラメータ解析用データ9を利用してそのコマンドを解析し、どのようなパラメータが指定されているか等を認識する。

パラメータ解析手段3による認識結果は、パラメータ変換手段4に与えられ、パラメータ変換手段4は、その認識結果とパラメータ変換用データ

段2と、パラメータ解析手段3と、パラメータ変換手段4と、エラー検出手段5と、エラー通知手段6と、コマンドプロシージャの書き込み手段7と、コマンドプロシージャの登録用ファイル13と、パラメータテーブル8と、パラメータ解析用データ9、パラメータ変換用データ10およびパラメータチェック用データ11を含むファイル14と、入出力処理を行なうための端末(端末装置)12とが示されている。ファイル14中のパラメータ解析用データ9は、パラメータ解析手段3によって利用されるデータであり、コマンドに指定されたパラメータの解析を行なうために必要な情報が格納され、パラメータ変換用データ10は、パラメータ変換手段4によって利用されるデータであり、コマンドに指定されたパラメータを処理内部で利用するパラメータテーブル8の形式に変換するための情報を持つ。また、パラメータチェック用データ11は、エラー検出手段5によって利用されるデータであり、各コマンド或いは機能に対して指定されるパラメータのエラーチェックに必要な各種の情報を

10を利用して、パラメータエラーチェック等に使用するパラメータテーブル8を作成する。

他方、作成者がフローチャート作成手段1を介してパラメータ入力手段2を呼出し、この呼出したパラメータ入力手段2によって端末12の画面に表示されるパラメータ入力画面に従ってコマンドのパラメータを指定した場合には、パラメータ入力手段2からどのようなパラメータが指定されたかがパラメータ変換手段4に通知され、パラメータ変換手段4は、その情報とパラメータ変換用データ10とに基づいてパラメータテーブル8を作成する。

さて、エラー検出手段5は、パラメータテーブル8を入力し、指定されたパラメータのチェックをパラメータチェック用データ11を利用して行なう。即ち、各パラメータに指定された値が、指定範囲、選択値、パラメータ相互の関連、指定位置等の点からみて適正であるかどうか、また必要なパラメータが揃っていないかどうか等をチェックする。

エラー検出手段5によってパラメータエラーが検出された場合、エラー通知手段6が呼出され、エラー通知手段6は、どのパラメータに、どんなエラーが検出されたか等の情報を端末12に出力し、コマンドプロシージャの作成作業を行なっている利用者に対してその訂正を促す。

作成者は、端末12の表示画面を見て誤りのあったパラメータを修正し、再度、フローチャート作成手段1若しくはパラメータ入力手段2を介して入力を行なう。

以上のような処理を繰り返すことにより、パラメータ指定エラーのないコマンドを指示していくことができる。

全てのコマンドの指定が終わり、フローチャートが完成したら、端末12からの指示によりフローチャート作成手段1はコマンドプロシージャの書き込み手段7を呼出し、複数のコマンド等から構成される完成したコマンドプロシージャを登録用ファイル13へ書込む。

以上が本実施例の概略動作であるが、本実施例

の理解を容易にするために、CALLコマンドの作成を例にして以下説明する。

CALLコマンドは、利用者プログラムのロードモジュールを実行するために使用するコマンドであり、例えば第2図に示すような形式を有しており、また各パラメータは以下のような内容と、パラメータ指定規則を有する。

①ファイル名

実行したいロードモジュールが格納されているロードモジュールライブラリの名前を指定するものであり、通常は必須のパラメータである。

②ロードモジュール名

実行したいロードモジュール名を指定する必須パラメータである。

③OPTIONS

実行したいプログラムに渡すオプション文字列を指定する。オプション文字列は300字以内であり、且つ前後をアポストロフィ(')で囲む。入力省略可能なパラメータである。

④CPTIME

RAWと指定し、何も指定されないときはアポートダンプの自動出力は行なわれないものと認識される入力省略可能なパラメータである。

さて、作成者が第1図の端末12からフローチャート作成手段1を利用してコマンドプロシージャの作成処理を行なっている場合であって、コマンドCALLの作成に際し、作成者がその機能に対応したパラメータ入力画面をパラメータ入力手段2から端末12の画面に呼出した場合、画面には例えば第3図に示すような内容が表示される。同図において、エリアa～fは各パラメータの入力エリアであり、aはファイル名、bはロードモジュール名、cはCPTIME、dはOUTLIM、eはDUMP、fはOPTIONSのエリアである。

作成者が上記表示されたコマンドCALLのパラメータ入力画面に従って各パラメータを入力していき、入力完了と思った時点で所定のキーを押下することにより、パラメータ入力画面に設定された内容がパラメータ入力手段2を介してパラメ

実行プログラムの使用CPU時間の制限値を指定する。指定は秒単位で整数6桁以内で指定する。無制限の場合は"NO"(NOLIM)と指定し、指定されない場合は所定の値が使用される入力省略可能なパラメータである。

⑤OUTLIM

実行プログラムの標準シスアウトへの出力制限値を指定する。指定は出力制限論理レコード数を整数9桁以内で指定し、無制限の場合は"NO"(NOLIM)と指定し、何も指定されないときは所定の値が使用される入力省略可能なパラメータである。

⑥DUMP

利用者プログラムが例外によって異常終了した場合のアポートダンプの自動出力の有無およびダンプの種類について指定する。ダンプをとらないときはNODUMP、書き込み可能なセグメントのダンプをとるときはDUMP=DATA、全セグメントのダンプをとるときはDUMP=ALL、メモリーイメージでダンプをとるときはDUMP=

ータ変換手段4に送られる。なお、パラメータ入力画面を用いてパラメータを指定する場合、パラメータ入力画面を用いずにパラメータを指定する方法に比べて誤りは少なくなるが、それでも必須パラメータの指定漏れや、パラメータ命名規則に違反した指定などは起こり易いものである。

他方、作成者がコマンドCALLの作成に際し、パラメータ入力画面を使用しない場合、フローチャート上に直接にパラメータを記述することになる。第5図は作成者が直接に記述したコマンドCALLに対応するパラメータ例を示す図であり、ファイル名としてLMLIBを、ロードモジュール名としてLMを、OPTIONSとして「…」を、OUTLIMとして500を、DUMPとしてNODUMPをそれぞれ指定し、CPTIMEは省略した例を示す。また、作成者の誤りによりコマンドCALLには定義されていないABCの内容を持つパラメータを入力した例が示されている。作成者は、必要なパラメータを入力し終えたと思った時点で所定のキーを押下すると、コマン

ドCALLに関する第5図の入力情報はフローチャート作成手段1を介してパラメータ解析手段3に送られ、解析される。

第4図はパラメータ解析用データ9、パラメータ変換用データ10及びパラメータチェック用データ11の内容例を示し、コマンドCALLに関する部分を抜き出したものである。同図においては、コマンド名CALLに対応して、機能名(コール)と第2図で説明した複数のパラメータが登録され、各パラメータについて内部処理で利用するナンバー、規定値の有無およびその値、キーワード(以上は主にパラメータ解析用データ9に属する)、パラメータ入力手段2による入力パラメータとの関連すなわち各パラメータは第3図のどの入力エリアに対応するか及びどのような表示方法が採用されているのか等の情報(これは主にパラメータ変換用データ10に属する)、必須パラメータか入力省略可能パラメータかの情報および指定文字数等のデータ属性、選択肢、パラメータ間の必須あるいは排他関係、パラメータ指定値間の必須ある

いは排他関係(以上は主にパラメータチェック用データ11に属する)がそれぞれ登録されている。

第1図のパラメータ解析手段3は、フローチャート作成手段1からコマンドCALLに関する入力情報として、例えば第5図に示すような内容を受取ったとすると、例えば第6図に示すような内容をパラメータ変換手段4に通知する。すなわち、第5図によれば、作成者はファイル名、ロードモジュール名、OPTIONS、OUTLIM、NODUMP、ABCの各パラメータを指定しているので、そのうちパラメータ解析用データ9に登録されているパラメータについては対応するナンバーを付けて、またABCの如くコマンドCALLに定義されていないパラメータについてはエラーである旨を付けて通知し、また作成者が入力を省略したパラメータ(今の例ではナンバー4のCPTIME)についてはパラメータ解析用データ9中の規定値を通知する。つまり、パラメータ解析手段3は、恰も第3図のパラメータ入力画面を使用して作成者がパラメータ入力を行なったとき

に得られような情報と、更に不必要なパラメータが入力された場合にはその旨の情報とをパラメータ変換手段4に通知する働きをする。

次にパラメータ変換手段4は、第6図に示す内容とパラメータ変換用データ10とを使用して、例えば第7図に示すようなパラメータテーブル8を作成する。第7図において、パラメータテーブル8は、機能名欄80と、コマンド名欄81と、パラメータがパラメータ入力手段2を介して入力されたのか或いはパラメータ解析手段3を介して入力されたのかについてのパラメータ入力方式欄82と、パラメータ番号、端末表示情報、エラー情報、内容の各欄を有するパラメータ情報欄83とを含んでいる。そして、第6図のような内容がパラメータ解析手段3から通知された場合、第8図の機能名欄80には「コール」が、コマンド名欄81には「CALL」が、パラメータ入力方式欄82には「パラメータ解析手段3経由」が各々登録され、またパラメータ情報欄83には同図に示すような内容が登録される。ここで、ナンバー6の端末表示

情報に「1」が設定されているのは、後述するようにエラーメッセージの表示時には第3図に示したパラメータ入力画面と同様な形式の画面を端末に表示するため、DUMPとしてNODUMPが指定されたとき表示画面のエリアに「1」を表示しなければならない為であり、これは第4図のパラメータ入力手段2からの入力パラメータとの関係情報から求めることができる。また、パラメータ情報欄83のナンバーが付されたパラメータの順番は、第4図のナンバーとパラメータ入力手段2からの入力パラメータとの関係情報との対応関係から、第3図のエリアで順に整えられている。更に、「ABC」の如き未定義なパラメータについては、エリア情報中に「未定義パラメータエラー」等の情報が格納される。

第1図のエラー検出手段5は、パラメータチェック用データ11を用いて、第7図のパラメータテーブル8中に登録された各パラメータのチェックを行なう。つまり、第4図に示すデータ属性、選択肢、パラメータ間の必須/排他関係情報、指定

値間の必須/排他関係情報を用いてチェックを行なう。例えば、必須パラメータであるファイル名やロードモジュール名の指定が抜けている場合にはその旨を第7図の対応するエラー情報欄に記述し、OPTIONSの指定文字列が300字を超えていればその旨を対応するエラー情報欄に記述するなどの処理を行なう。また、今例にしたコマンドCALLについては存在しないが、或るパラメータを指定したときには必ず或る別のパラメータを指定しなければならないのにその別のパラメータの指定がなければ、その旨を対応するエラー情報欄に登録する等の処理を行なう。

パラメータ検出手段5の上述した処理が終了し、パラメータエラーが検出されると、パラメータテーブル8の内容がエラー通知手段6に通知され、エラー通知手段6はパラメータテーブル8の内容に基づいて、例えば第8図に示すように作成者が入力したパラメータの内容とエラーメッセージとを端末12の画面に表示し、作成者にパラメータ指定にエラーがある旨およびそのエラーの内容など

を通知する。これに応じて作成者はパラメータの設定をやり直すが、本実施例ではパラメータ入力手段2によるパラメータ入力画面と同様な内容が表示されているので、作成者が必要な入力を終了して所定のキーを押下すれば、自動的にパラメータ入力手段2が呼出され、このパラメータ入力手段2に上記パラメータ入力画面が引き渡されるように構成されており、入力操作のより一層の簡略化が図られている。

また、今回のコマンドについて何もエラーがなければ、エラー検出手段5からエラー無しの通知が例えばフローチャート作成手段1に為され、フローチャート作成手段1はパラメータテーブル8のうち必要な部分を、作成されたコマンドプロシージャの一つのコマンドに対応する情報として保持し、次のコマンドに処理を進める。

なお、上記実施例では、エラーメッセージを端末12に表示する際に、パラメータ入力手段2が使用するパラメータ入力画面と同様な形式の内容と、エラーメッセージとを合わせて表示したが、単に

エラーメッセージだけを表示するようにしても良い。

また、以上の実施例はコマンドプロシージャの作成について本発明を適用したが、ジョブ制御言語の作成についても同様に適用可能である。

(発明の効果)

以上説明したように、本発明は、コマンドやジョブ制御言語に指定されるパラメータを、その入力方式を問わず即ち端末に表示されたフローチャート上に直接記述した場合とパラメータ入力手段によるパラメータ入力画面から入力した場合ともに、そのコマンドやジョブ制御言語の作成時点でチェックし、若し誤りを検出すればエラーメッセージを端末に表示するようにしたので、作成者はコマンドやジョブ制御言語の作成段階でパラメータエラーを知ることができ、正しいパラメータを指定し直すことにより、エラーの少ないコマンドプロシージャやジョブ制御言語を登録用ファイルに登録することができる。この為、作成したコマンドプロシージャやジョブ制御言語を実行して初

めてエラーが判明する従来に比べ、コマンドプロシージャやジョブ制御言語の作成期間を短縮できると共に電子計算機システムの有効利用が可能となり、またコマンドやジョブ制御言語に指定するパラメータについて詳細な知識を持たない利用者でも適正なコマンドプロシージャ、ジョブ制御言語の作成ができ、作業の無駄も省くことができる。

4. 図面の簡単な説明

第1図は本発明の実施例のブロック図、

第2図はコマンドCALLのパラメータ形式例を示す図、

第3図はパラメータ入力手段2によるパラメータ入力画面の内容例を示す図、

第4図はパラメータ解析用データ9、パラメータ変換用データ10およびパラメータチェック用データ11の内容例を示す図、

第5図は利用者が直接に入力したパラメータ例を示す図、

第6図はパラメータ解析手段3の出力の内容例を示す図、

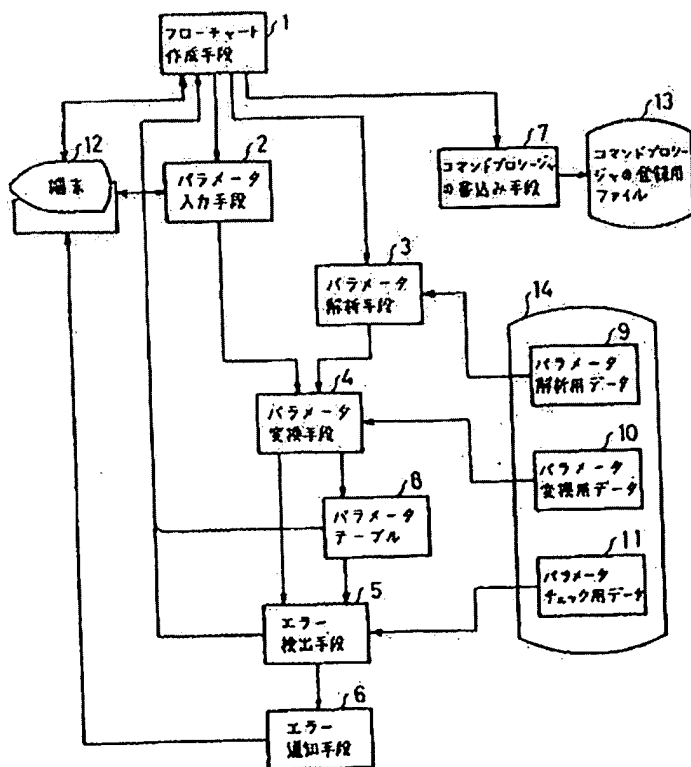
第7図はパラメータテーブル8の内容例を示す図および、

第8図はエラーメッセージが表示された画面の内容例を示す図である。

図において、1…フローチャート作成手段、2…パラメータ入力手段、3…パラメータ解析手段、4…パラメータ変換手段、5…エラー検出手段、6…エラー通知手段、7…コマンドプロシージャの書き込み手段、8…パラメータテーブル、9…パラメータ解析用データ、10…パラメータ変換用データ、11…パラメータチェック用データ、12…端末、13…コマンドプロシージャの登録用ファイル、14…ファイル。

特許出願人 日本電気株式会社

代理人 弁理士 境 廣 巳



本発明の実施例のブロック図

第1図

コマンド名	パラメータ
CALL	ファイル名(ロードモジュール名) (OPTIONS='オプション文字列') {CPTIME={CPU使用制限時間}} {OUTLIM={出力制限数}} {NODUMP={}} {DUMP={DATA, ALL, RAW}}

コマンドCALLのパラメータ形式例を示す図

第2図

----- ロードモジュールの実行 -----

パラメータ入力/確認

LMライブラリ名 ⇒ a

LM名 ⇒ b

CPU使用制限時間 ⇒ 秒 (NO: 無制限)

出力制限数 ⇒ 行 (NO: 無制限)

ダンプ採取 ⇒ ☐ c (1: とらぬ 2: データのみ 3: 全セグメント 4: メモリイメージ)

実行時オプション文字列 ⇒ d

パラメータ入力画面の内容例を示す図

第3図

		9			10		11			
コマンド名	機能名	パラメータ	ナンバー	既定値	キーワード	2による入力パラメータとの関連	データ属性	選 択 肢	パラメータ間の必須/排他	指定値間の必須
CALL	コール	ファイル名	1	無	無	a	必須,...	無	—	—
		ロードモジュール名	2	無	無	b	必須,...	無	—	—
		OPTIONS	3	無	"OPTIONS"	f	省略可,...	無	—	—
		CPTIME	4	NO	"CPTIME"	c	省略可,...	無	—	—
		OUTLIM	5	NO	"OUTLIM"	d	省略可,...	無	—	—
		DUMP	6	無	"NODUMP" "DUMP"	e 1..., 3..., 2..., 4...,	省略可,...	NODUMP DUMP { DATA ALL RAW	—	—

パラメータ解析用データ9, パラメータ変換用データ10および
パラメータチェック用データ11の内容例を示す図

第 4 図

```
CALL LMLIB (LM)
OPTIONS = '.....'
OUTLIM = 500
NODUMP
ABC
```

利用者が直接に入力したパラメータ例を示す図

第 5 図

CALL	
パラメータのナンバー	指 定 値
1	LMLIB
2	LM
3	"....."
5	500
6	NO
4	NO
なし (ABC)	未定義パラメータエラー

パラメータ解析手段3の出力内容例を示す図

第 6 図

80	コール			
81	CALL			
82	パラメータ解析手段経由			
83	パラメータ情報			
	ナンバー	端末表示情報	エラー情報	内容
	1	LMLIB		LMLIB
	2	LM		LM
	4	NO		NO
	5	500		500
	6	1		NO
	3	...		'...'
		ABC	未定義パラメータエラー	ABC

パラメータテーブル8の内容例を示す図

第 7 図

----- ロードモジュールの実行 -----

LMライブラリ名 ⇒ LMLIB

LM名 ⇒

CPU使用制限時間 ⇒ NO 秒 (NO: 無制限)

出力制限数 ⇒ 500 行 (NO: 無制限)

ダンプ採取 ⇒ 1 (1: どちらもない 2: データのみ
3: 全セグメント 4: メモリイメージ)

実行時オプション
文字列 ⇒

エラーメッセージ ABC: 未定義パラメータ OPTIONS: 文字制限オーバー
LM名: 未入力

エラーメッセージが表示された画面の内容例を示す図

第 8 図